



**QUEEN'S  
UNIVERSITY  
BELFAST**

## A Smart Office System with Face Detection at the Edge

Prentice, C. T. J., & Karakonstantis, G. (2018). A Smart Office System with Face Detection at the Edge. In *2018 IEEE Smart World Congress: Proceedings* (2018 ed., pp. 88-93). Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/SmartWorld.2018.00050>

**Published in:**  
2018 IEEE Smart World Congress: Proceedings

**Document Version:**  
Publisher's PDF, also known as Version of record

**Queen's University Belfast - Research Portal:**  
[Link to publication record in Queen's University Belfast Research Portal](#)

**Publisher rights**  
© 2018 The Authors.  
This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

**General rights**  
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**  
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [openaccess@qub.ac.uk](mailto:openaccess@qub.ac.uk).

## A Smart Office System with Face Detection at the Edge

Prentice, C. T. J., & Karakonstantis, G. (Accepted/In press). A Smart Office System with Face Detection at the Edge. In 2018 IEEE Smart World Congress: Proceedings (2018 ed., pp. 88-93). IEEE .  
<https://doi.org/10.1109/SmartWorld.2018.00050>

**Published in:**  
2018 IEEE Smart World Congress: Proceedings

**Document Version:**  
Publisher's PDF, also known as Version of record

**Queen's University Belfast - Research Portal:**  
[Link to publication record in Queen's University Belfast Research Portal](#)

### **General rights**

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

### **Take down policy**

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [openaccess@qub.ac.uk](mailto:openaccess@qub.ac.uk).

## A Smart Office System with Face Detection at the Edge

Christopher T. J. Prentice

*School of Electronics, Electrical Engineering and Computer  
Science,*  
Queen's University Belfast.  
United Kingdom  
[cprentice01@qub.ac.uk](mailto:cprentice01@qub.ac.uk)

Georgios Karakonstantis

*School of Electronics, Electrical Engineering and Computer  
Science,*  
Queen's University Belfast.  
United Kingdom  
[g.karakonstantis@qub.ac.uk](mailto:g.karakonstantis@qub.ac.uk)

**Abstract**— The rapid increase in the number of Internet connected devices has placed a high level of demand on both the network bandwidth and the processing power currently available to the centralized physical datacentres that embody the ‘Cloud’. To alleviate such a demand and relieve the network capacity, the Edge Computing paradigm was recently proposed. Such a paradigm evangelizes the processing of data locally, closer to users and so avoiding the slow communication of data to centralized datacentres. The realization of such a paradigm requires the evaluation of smart applications on various potential energy efficient devices by understanding their processing and storage limits, while also looking for efficient methods to improve their capabilities. In this paper, we develop and evaluate an end to end smart office application on a representative Edge device, the latest Raspberry-Pi, while utilizing existing Cloud on Demand services connected through an Android application and Amazon Alexa Skill. The developed solution monitors various environmental conditions and is able to recognize users using facial recognition. We will evaluate the requirements of such an application on an Edge device and explore methods to reduce the data stored and processed, while evaluating the impact this has on the detection rate. In particular, we evaluate the effectiveness of increasing the image compression levels by measuring the level of compression used versus the time to create trained data and the face detection rate it produces. Our experimentation shows favourable results for the Edge system implementation, while also supporting the possibility of the hybridisation of Cloud and local processing to achieve complex tasks while minimising network use.

**Keywords**— *IoT, Facial Recognition, Edge Computing, Smart Environment*

### I. INTRODUCTION

The growth of the ‘Internet of Things’ (IoT), and the markets surrounding the development and sale of the devices that it represents has been rapid. Upper estimations have indicated that there will be more than 75 billion devices connected to the internet by 2025, creating a market that will be worth more than one billion dollars annually [1]. This increase in the IoT field has meant that the ability for these devices to monitor and control the surrounding environments has increased exponentially. Exemplified by the ever-growing trend of smart appliances and smart homes/environments, this boom in smart applications has created vast amounts of new data that has applications for both domestic use and wider

research and development projects. However, the sheer amount of data being produced has put a lot of pressure on the current Cloud infrastructure. While it is also expanding at a rapid rate [2], there are now doubts if the current physical Datacentre infrastructure will be able to accept, process and return the data being produced within the required timeframes [3]. In fact, it is expected that the amount of data traffic being produced by IoT devices will reach 24.3 exabytes in 2019 putting a huge strain on the current Infrastructure [4]. Therefore, a new paradigm is needed to allow the efficient handling, processing and storage of the data being produced by various applications, especially the security ones where response time is crucial.

#### A. Reducing the Load on the Cloud

There are two concepts currently being researched in an effort to effectively reduce the load being placed on the Cloud and increase the performance of IoT systems. These concepts are commonly referred to as Fog and Edge Computing and while there are similarities between the two, there is a difference in the execution of their final systems.

Fog Computing, a term first coined by Cisco [5], is a system that would focus on one high power gateway in a local area network of IoT device. This gateway system would gather, process, and store the data from the local devices. Once received and processed the pertinent data and any addition commands or instructions would then be transmitted back out from the gateway to the necessary IoT devices or external systems.

Edge Computing, on the other hand, evangelizes bringing Cloud services even closer to the end users than Fog Computing and removing some of the complexity that a Fog implantation would entail. Carrying out as much data processing on the IoT device that is recording/creating the data before then going to any wider network even the local LAN [6]. This ties into the belief that it would be more cost and energy efficient to either entirely remove or severely cut down on the network traffic that an IoT system requires and the load it creates in physical datacentres. With local devices acting independently for simple to medium tasks and only using networked resources for more advanced and complex problems.

With the simplicity and ease of deployment of Edge Computing, it is the logical choice in small-scale projects similar to the previously mentioned smart environments. Supported by energy efficient devices that collect data and process it locally rather than requiring the transport of this information to expensive high-end servers or central network gateways, the system reduces both cost of production and latency between sensor/data input and meaningful output from the system. However, modern computing hardware has not reached a sufficient level where all data processing can occur fast enough that the network induced latency is not the preferable option in some situations.

There is still a need to investigate energy efficient methods for porting Edge applications to such IoT devices. The reduction in time and energy needed to locally process data may then make it worth forgoing the forwarding of data on to a Cloud system. Alternatively, it may support the creation of a hybrid solution, where pre-processing of the raw data would be carried out locally before sending on the smaller processed data packets to the aforementioned Cloud facilities.

### B. Contributions

Within this paper the application of facial recognition and environment monitoring within an IoT environment is utilised as a case study for evaluating this new paradigm and its effectiveness within real world scenarios and implementations.

An end to end Smart Office system was implemented that utilised local sensors and a connected web camera to simultaneously record environmental information and carry out facial recognition of any user. Amazon's Alexa Cloud service was also utilized to create a more complete user interface while also enabling the exploration of a realistic Cloud-Edge interaction. A custom Alexa Skill was implemented to enable the reception and transmission of data to and from the Cloud service to the Edge device/system for fulfilling user requests.

To evaluate one of the changes that could be made to reduce the amount of storage and processing needed to deal with IoT data, the level of compression utilised for storing images used for facial recognition was varied. By doing so we explored the relationship between the compression level and the produced stored image data, with the accuracy of the training data that it would produce. We have also implemented a result smoothing algorithm which was found to be able to improve the prediction accuracy with a minor increase in processing. Our results indicated that a purely Edge-based system or a hybrid one utilizing Edge and Cloud components could be supported by the Edge based device and so help minimize the use of the network and so alleviate the expected flood of data.

The paper is organized as follows. Section II discusses the Smart Office system created, including its hardware, software and more detail on Facial Recognition. Section III elaborates on the Alexa Digital Assistant giving background regarding its standard execution and an explanation as to how it is a standard Cloud-based computing implementation. The evaluation of the system created is then carried out in Section IV, using the defined metrics to analyse the effects on compression levels on facial recognition and its required processing. The effects of

Edge Computing and the different implementations that could be utilised are outlined in Section V. Finally, conclusions are drawn in Section VI.

## II. RASPBERRY PI BASED SMART OFFICE

To evaluate some of the factors that have been identified in this paper, a system to create a smart office environment that included the data intensive task of facial recognition was developed, focusing on local processing power to achieve the outlined development requirements. However, during development a Cloud-based message broker system was utilised to allow for communication between individual components. Figure 1 depicts an outline of the implemented end to end system.

### A. Smart Office Hardware

A Raspberry Pi 3 was selected to be the core of this system due to its balance between performance and form factor (technical specifications outlined in Figure 2), as well as its symbolism within the IoT community. With the Arduino microcontroller (2005) [7] and Raspberry Pi SBC (2012) [8] being a well-contributed factor into the growth of the 'Maker' community that has itself impacted the commercial and research spaces of the IoT domain [9]. Alongside the Pi, due to the systems open design, a GrovePi development kit (with light, temperature and humidity sensors) from Seeed Studios and a standard Logitech web camera were selected to serve as the I/O devices of the system.

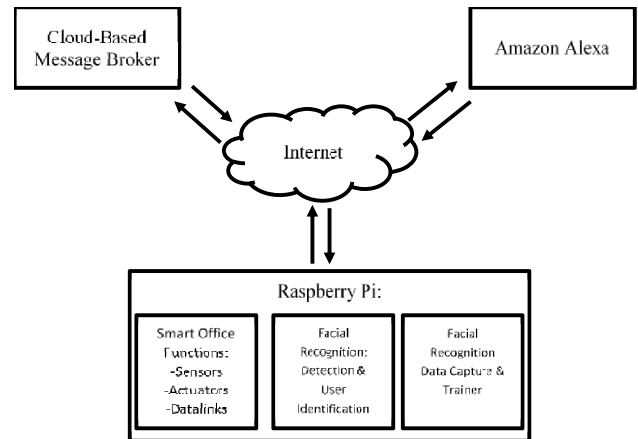


Figure 1. - Diagram outlining system components

Category	Specification
Processor	Quad Core 1.2GHz Broadcom BCM2837 64-bit CPU
Memory	1GB RAM
Wireless Connectivity	BCM43438 wireless LAN and Bluetooth Low Energy (BLE)

Storage	Micro SD port for loading your operating system and storing data
Display	Full size HDMI out
Power	Upgraded Micro USB power source up to 2.5A
Various	40-pin extended GPIO
	4 USB 2 ports
	CSI camera port for connecting a Raspberry Pi camera

Figure 2. - Hardware Specification of Raspberry Pi 3

### B. Smart Office Application

Computer Vision in general, and the sub-topic of Facial recognition specifically, is an expanding field of research. This is evident with the many commercial and academic projects and products that have been centred around the application of this technology. In recent years there have been a series of developments in the utilisation of machine intelligence and deep learning to improve these systems and make them more easily accessible to the general populous. One such popular project, as of writing, is the application of Google's Tensor flow with the YOLO real-time object detection and classification network [12], with various teams and individual developers looking at its applications and abilities. In our application, to reduce the amount of processing power of the Pi that would need to be dedicated to image recognition, this project will not go as far as implementing a neural network based system, such as TensorFlow but shall instead manage image processing as well as facial recognition with the utilisation of the OpenCV v2.4 library.

There are well documented studies on OpenCV's abilities to function on the Raspberry Pi platform [13]. With the use of the OpenCV library, the project is able to rely on OpenCV's pre-generated Haar Cascade classifier, these classifiers could have been produced during the project [14] but with the wide use of OpenCV, the pre-packaged data has been proven to be reliable and so was utilised to reduce time rehashing previous works. Once a face has been detected in the video feed there are then three main algorithms that are commonly used to carry out facial recognition: Eigen Face, Fisher Face and Local Binary Pattern Histogram (LBPH). Through research into previous studies of these algorithms, it has been found that LBPH has the recommendation for use, due to its efficiency and accuracy [15] and so, therefore, it shall be the Algorithm

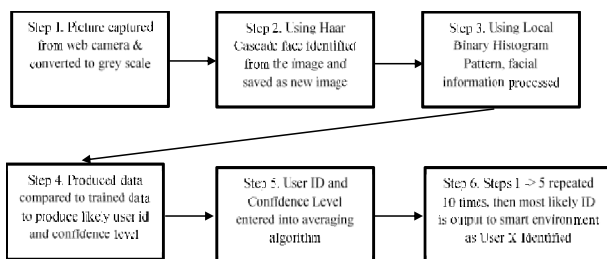


Figure 3. - Flow Diagram of Facial Recognition Functionality

utilised during the experimentation with the system. This accuracy and an output smoothing method to improve its results was also tested during the course of projects development as one facet of the goal of demonstrating the viability of local lower powered processing.

The smoothing algorithm worked on the principle that multiple images of a user could be taken as they enter the camera's field of view. With this collection of images, each could be individually evaluated before those evaluations are compiled into one final decision upon the user's identification. To implement this process the following steps were followed:

1. A number of images would be captured when a face is detected by the Haar Cascade.
2. These images would then pass through the LBPH and the results (consisting of an ID and confidence value) of each image being recorded. If the same ID appeared more than once then its confidence values would be grouped together.
3. If one ID was identified the most, then it would be selected as the output result.
4. However, if there were multiple IDs with the same number of identifications then each of their confidence levels would be examined and the ID with the highest confidence would then be output.

For example:

If users A, B and C are registered to use the system and the camera is triggered, producing the following results:

A (57), B (87,83,98,89,89,97,86), C (62,67)

Then B shall be the final result as it has the majority in the output results: 7/10. However, if the initial results were:

A (34,45,12,31), B (21,15), C (83,98,89,97)

Then C shall be the final result as even though both A and C have 4 out of a possible 10, C has the highest confidence level.

### III. INTEGRATION WITH ALEXA DIGITAL ASSISTANT

During the design of the system, the inclusion of a connection to the Alexa Digital Assistant Service was included to provide a natural user interface, where end-users would be able to make various queries and requests to the system.

#### Alexa Skills Kit: Processing a request



Figure 4. - Diagram of Amazon Alexa Service [11]

While not having an effect on the current experiment, ‘Alexa’ is an ideal example of pure Cloud-based computing, where the local system only records audio clips that are then sent on to Amazon servers to be processed by their Cloud services and execute programs and scripts located on the Cloud through the use of webservices. [10]

During the implementation of the required ‘Alexa Skill’, Amazon’s name for Alexa applications, the Cloud service reliably completed the required tasks within 300ms of receiving the request. Yet depending on network traffic and the location of the data centre being utilised, a delay of 1 to 5 seconds could be observed sporadically in the system. Though without further access and research these findings will not be included in the scope of the investigation covered in this paper.

#### IV. EVALUATION

Once we have validated the correct functionality of our system and of all the sensors, we have evaluated it by focusing on the face detection application which is the most demanding in terms of data processing and storage. Note that when we were performing our evaluations all the mentioned sensors in Section II were active and reporting their results.

##### A. Face detection trade-offs

To optimize the face detection processing carried out on the Pi, the impact of using different image compression levels was evaluated. This compression can help reduce the amount of data that needs to be stored and processed. The smoothing algorithm previously mentioned in Section II was also included to see if data smoothing could improve the accuracy of the system while not requiring a large amount of additional processing. The following metrics were used to analyse the results produced by the system:

- The accuracy of each prediction made
- The accuracy of the smoothing algorithm
- The time taken to process these images into the trained data set
- Difference in power usage of the Raspberry Pi during this processing.

To carry out this experiment it was decided that the use of a test set of 10 users would be suitable, with 15 images of each candidate’s face required. This would allow for a broad range of users, with differences in gender, features and race, while also providing 10 images to be used for training the system and 5 images to be used to assess its performance. To fulfil this requirement 10 individuals with the required 15 pictures were selected randomly from the MUCT Face Database [16] as this data was freely available and has been used previously in other research papers, including the original paper that produced these images [17].

Each of the user images used in the training function were formatted so that they would represent ten distinct levels of image quality, while the remaining tester images retained in a controlled high-quality PNG format, to remain as a controlled variable. These different image levels were then used to

produce trained recognition data to be passed onto the recognition script that would be used to verify the accuracy of each the different levels of detail. See the following Figures 5 through 8 for graphs of the results obtained: -Note that in the ‘Format’ axis, JPEG 20% stands for an image in JPEG format at 20% the file size of the original JPEG image.

With the experimental data produced by the system testing, it became clear that the Raspberry Pi was indeed capable of carrying out facial recognition locally with no external assistance. Additionally, the data produced did not meet the initial expectation that a higher image quality used during training would lead to a higher number of correct user identifications.

While there was an initial increase in accuracy, once file sizes reached 50% of the size of the initial JPEG there were no gains in accuracy. In fact, the accuracy of the standard OpenCV algorithm reached peak performance at 60% original file size, as thereafter the system started to return the same false positive identification for one image, regardless of the

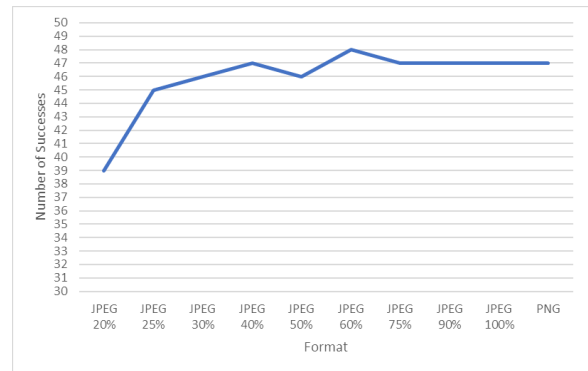


Figure 5. - Line graph displaying the variation in the number of successful recognitions against the format used during training. Y scale has been changed from 0-50 to 30-50 to display these differences more clearly.

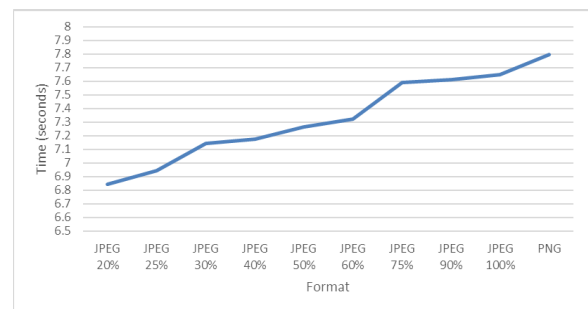


Figure 6 – Line graph illustrating the differences in time for execution of the Training algorithm for each of the supplied formats. With a Max value of 7.8 seconds for PNG format and a Min value of 6.85 for the JPEG 20% format, we can observe less than a 1 second increase in processing time between best and worst-case scenarios. Though this represents a 14% increase in execution time, the file size’s increase is more in the range of 820%.

level of detail utilised. This created an interesting dynamic between image quality and accuracy but on further research, other papers have produced proof that image quality does not affect the recognition algorithm as much as was first thought. The authors of the paper “Image Compression Effects on Face Recognition for Images with Reduction in Size” [18] researched the effects of file size reduction and while utilising Eigen Faces as their identification algorithm rather than LBPH, the results of their experimentation still showed that accuracy only showed degradation at under 10% of the initial size.

In attempting to calculate power usage, it became clear that the power used in processing the images was directly proportional to the time taken to do so. Due to the speed that the process is completed in and the fact that each of these results was within a range of 6.8-7.8 seconds, the power results showed little to no variance and in fact, stood within the accuracy tolerances of the measuring power meter. This was most likely due to the fact, that when the training data was being processed the Raspberry Pi’s processor would show 100% load on a single processor. While a larger data set may have been able to more clearly display the power differences, if we take the standard 100% usage on one core as a constant power draw during the training data creation, then the power levels will be reflected in the time to process.

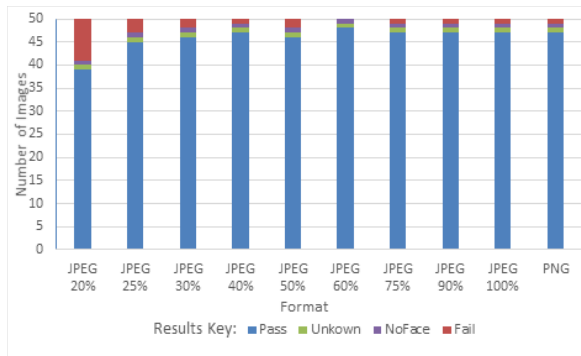


Figure 7 – Bar chart displaying the raw results of all 50 images from the 10 sets of user data against the different formats that were used during training.

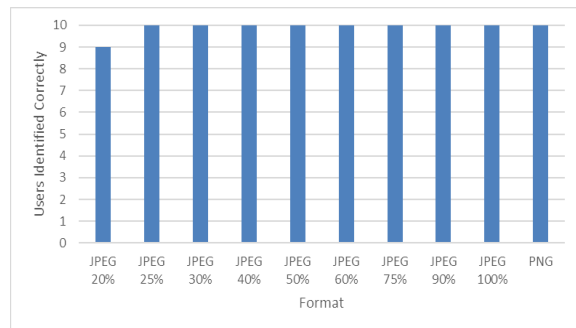


Figure 8 - Bar chart showing the number of times that the stabilisation algorithm was correct for the ten sets of users in the system, against the different formats that were used during training.

In regard to the smoothing algorithm that was developed for the project, Figure 8 proves it successfully increased the accuracy of the predictions being made by the system. It succeeded in correctly identifying users for all levels of detail bar when file sizes were at their lowest of 10% of the original value and the base facial recognition function accuracy percentage dropped below 80%. With this clear advantage over the standard recognition functionality, the stabilisation algorithm was a success and achieved its goal of producing more reliable results without drastically increasing processing requirements.

## V. EVALUATION OF POSSIBLE IMPROVEMENTS THAT COULD BE MADE WITH THE INCLUSION OF CLOUD COMPUTING

With the results produced by the previous experimentation, it became clear that the use of Cloud Computing would only affect the creation of training data in a significant manner. While the time saved in processing versus the size of the files used is not linear and so the local system would be fine for a smaller number of users, it is interesting in a point that Cloud Computing could speed up the progress due to the increase of raw computing power rather than the image size being used. Therefore, the images could be compressed and sent to the Cloud to leverage its improved processing capabilities without a loss in accuracy of the predictions made. With this in mind four separate possibilities were identified which varied in the amount of Cloud Computing that would be introduced into the Edge Computing system.

The first possibility is the simplest, in that no Cloud Computing would be used. The total time required for the system was reasonable and while storing images would eventually prove difficult for the Raspberry Pi, with consumer micro SD memory cards easily reaching 128GB, it is still a viable and cheaper alternative that would be attractive for low cost, low work load system.

The next option would be to offload the creation of trained recognition data to the Cloud. Here once the sample images needed for its creation are captured on the local system, they would be sent to the Cloud where a training algorithm would create then return the required data for the local system to then use in facial recognition. This option would reduce storage needs on the local device, as well as speeding up the slowest part of the current facial recognition system. Due to the use of Cloud Computing the processing power could be dynamically scaled to the numbers being added to the system and so would support a wider implementation than the previous option.

Leading on from this the next possibility, which would be to continue using the Haar Cascade classifier on the Raspberry Pi to identify images with faces in them, but then instead of processing these images locally and attempting user identification, the system would send the specified images to the Cloud, which would then use its own stored recognition data to carry out the required processing and identification before then passing back the identity of the user in the image.

The final and most drastic option is the complete offloading of the facial recognition component of the system to the Cloud. This would completely remove the Edge Computing aspect of the project. If this was to make use of Machine Learning or



another form of processing beyond the Raspberry Pi then it could be argued that it is worth the additional latency and costs. However, it is doubtful that any other reasoning would nullify the adverse effects caused by the amount of data traffic and power that would be needed to implement this option. This is even more poignant if the Smart Office system had multiple instances, as this would create several feeds that the Cloud infrastructure would have to receive and then process fast enough to return if a user is detected in the feed to the correct IoT system.

## VI. CONCLUSIONS

While the current system remains a test bed for various implementation methods, this paper outlines the initial results that have been produced by a study of the Raspberry Pi as an Edge IoT Device capable of carrying out the challenging task of facial recognition, alongside the standard smart environment tasks of monitoring various sensors and actuators. While Cloud Computing facilities could be utilised in the processing of training data and carry out the actual recognition task, it has been shown that the time taken to carry out the facial recognition locally is acceptable. Therefore, the possible latency from data transmission would mean that the timings are comparable, even when more powerful Cloud servers are utilised. With this in mind, as well as the fact that a local system implementation will mean that once training data is loaded then there will be no dependencies on a reliable internet connection. Then it is advisable that either a fully local system or the second option of a hybrid system would be the most advantageous for our continued research into the implementation of Facial recognition within a Smart Office environment.

## VII. ACKNOWLEDGEMENTS

The work presented was partially supported by the UniServer project under the EU Horizon 2020 programme and grant no. 688540. We thank, S. Milborrow, J. Morkel and F. Nicolls, authors of the MUCT Database utilised during testing and experimentation.

## VIII. REFERENCES

- [1] ISH Markit Ltd, "IoT installed base, global, market, billions", statista.com, November 2016. [Online], Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide> [Accessed: May. 12, 2018]
- [2] Cisco, "Cisco Global Cloud Index Projects Cloud Traffic to Nearly Quadruple Representing 92 Percent of Total Data Center Traffic by 2020", November 2016. [Online], Available: <https://newsroom.cisco.com/press-release-content?articleId=1804748> [Accessed: May 12, 2018]
- [3] Y. Sverdlik, "Gartner: internet of things will disrupt the data center", March 2014. [Online], Available: <http://www.datacenterdynamics.com/content-tracks/servers-storage/gartner-internet-of-things-will-disrupt-the-data-center/85635.fullarticle> [Accessed: May 12, 2018]
- [4] Cisco, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update", March 2017. [Online], Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html> [Accessed: May 20, 2018]
- [5] M. Abdelshkour, "IoT, from Cloud to Fog Computing", March 2015. [Online], Available: <https://blogs.cisco.com/perspectives/iot-from-Cloud-to-fog-computing> [Accessed: May 20, 2018]
- [6] M. Satyanarayanan, "Edge Computing", Computer, vol. 50, issue 10, Oct. 2017.
- [7] D. Kushner, "The Making of Arduino", IEE Spectrum, 26 Oct 2011. [Online], Available: <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino> [Accessed: May. 12, 2018]
- [8] L. Upton, "We've Started Manufacture!", raspberrypi.org, 10 Jan 2012. [Online], Available: <https://www.raspberrypi.org/blog/weve-started-manufacture/> [Accessed: May. 12, 2018]
- [9] National Instruments, "The World Is Ours to Make: The Impact of the Maker Movement". [Online] Available: [http://www.ni.com/pdf/company/en/Trend\\_Watch\\_Maker.pdf](http://www.ni.com/pdf/company/en/Trend_Watch_Maker.pdf) [Accessed: May. 14, 2018]
- [10] Amazon, "Understand Custom Skills", developer.amazon.com. [Online], Available: <https://developer.amazon.com/docs/custom-skills/understanding-custom-skills.html> [Accessed: May. 20, 2018]
- [11] Amazon Web Services, "Creating IoT Solutions with Serverless Architecture & Alexa", Feb 2017. [Online], Available: <https://www.slideshare.net/AmazonWebServices/creating-iot-solutions-with-serverless-architecture-alexa> [Accessed: May. 20, 2018]
- [12] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", arXiv.org, 9 May 2016. [Online], Available: <https://pjreddie.com/media/files/papers/yolo.pdf> [Accessed: May. 20, 2018]
- [13] S. C. Gaddam, N. V. K. Ramesh and H. Dhanekula, "Face Recognition Based Attendance Management System with Raspberry Pi 2 using Eigen Faces Algorithm", ARPN Journal of Engineering and Applied Sciences, vol. 11, no.13, Jul 2016.
- [14] S. Soo, "Object detection using Haar-cascade Classifier", University of Tartu. [Online], Available: <https://pdfs.semanticscholar.org/0f1e/866c3acb8a10f96b432e86f8a61be5eb6799.pdf> [Accessed: Jun. 10, 2018]
- [15] S. Narang, K. Jain, M. Saxena and A. Arora, "Comparison of Face Recognition Algorithms Using OpenCv for Attendance System", International Journal of Scientific and Research Publications, vol 8, issue 2, Feb 2018.
- [16] S. Milborrow, "The MUCT Face Database", Oct. 27, 2015. [Online] Available: <https://github.com/StephenMilborrow/muct> [Accessed: Jun. 16, 2018]
- [17] S. Milborrow, J. Morkel and F. Nicolls, "The MUCT Landmarked Face Database", Pattern Recognition Association of South Africa, 2010. Available: <http://www.milbo.org/muct> [Accessed: Jun. 10, 2018]
- [18] V.K. Padmaja, G. Prasad, B. Chandrasekhar, "Image Compression Effects on Face Recognition for Images with Reduction in Size", International Journal of Computer Applications, vol. 61, no. 22 Jan 2013.